



Integration of Embedded Data Processing Algorithms Inside PAMELA Devices

Mariano Ruiz, Eduardo Barrera, Nestor Fernández, Raul Meléndez, Angel Alcaide, Gerardo Aranguren, Pedro María Monje

► To cite this version:

Mariano Ruiz, Eduardo Barrera, Nestor Fernández, Raul Meléndez, Angel Alcaide, et al.. Integration of Embedded Data Processing Algorithms Inside PAMELA Devices. Le Cam, Vincent and Mevel, Laurent and Schoefs, Franck. EWSHM - 7th European Workshop on Structural Health Monitoring, Jul 2014, Nantes, France. <hal-01020304>

HAL Id: hal-01020304

<https://hal.inria.fr/hal-01020304>

Submitted on 8 Jul 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

INTEGRATION OF EMBEDDED DATA PROCESSING ALGORITHMS INSIDE PAMELA DEVICES

Mariano Ruiz¹, Eduardo Barrera¹, Nestor Fernández¹, Raul Meléndez¹, Angel Alcaide²
Gerardo Aranguren³, Pedro María Monje³.

¹ *Instrumentation and Applied Acoustic Research Group. Technical University of Madrid (UPM),
Madrid, Spain*

² *AERnova Engineering Solutions Ibérica S.A, Madrid, Spain*

³ *Electronics Design Group. University of the Basque Country, Bilbao, Spain*

eduardo.barrera@upm.es

ABSTRACT

PAMELA (*Phased Array Monitoring for Enhanced Life Assessment*) SHMTM System is an integrated embedded ultrasonic guided waves based system consisting of several electronic devices and one system manager controller. The data collected by all PAMELA devices in the system must be transmitted to the controller, who will be responsible for carrying out the advanced signal processing to obtain SHM maps. PAMELA devices consist of hardware based on a Virtex 5 FPGA with a PowerPC 440 running an embedded Linux distribution. Therefore, PAMELA devices, in addition to the capability of performing tests and transmitting the collected data to the controller, have the capability of perform local data processing or pre-processing (reduction, normalization, pattern recognition, feature extraction, etc.). Local data processing decreases the data traffic over the network and allows CPU load of the external computer to be reduced. Even it is possible that PAMELA devices are running autonomously performing scheduled tests, and only communicates with the controller in case of detection of structural damages or when programmed. Each PAMELA device integrates a software management application (SMA) that allows to the developer downloading his own algorithm code and adding the new data processing algorithm to the device. The development of the SMA is done in a virtual machine with an Ubuntu Linux distribution including all necessary software tools to perform the entire cycle of development. Eclipse IDE (Integrated Development Environment) is used to develop the SMA project and to write the code of each data processing algorithm. This paper presents the developed software architecture and describes the necessary steps to add new data processing algorithms to SMA in order to increase the processing capabilities of PAMELA devices. An example of basic damage index estimation using delay and sum algorithm is provided.

KEYWORDS : *structural health monitoring, Lamb waves, piezoelectric transducers, embedded data processing.*

INTRODUCTION

Structural Health Monitoring (SHM) systems imply the integration of non-destructive evaluation methods within a structure to enable autonomous state awareness for structural-integrity. SHM systems offer methods to reduce operational costs and to improve the safety and reliability of aircrafts and other structures. The SHM approach can provide continuous monitoring, inspection, and detection of damage in structures with minimal human involvement. Ultrasonic inspection methods are based on Lamb waves. This elastic waves can be generated by applying excitation

signals to piezoelectric transducers. The ultrasonic waves propagate throughout the structure, interact with obstacles, flaws and borders in the structure, and reflect back to the transducers. The piezoelectric transducers can acquire the reflected signals, containing information about the dimensions of the structure and the obstacles found that modify the trajectory of the waves.

PAMELA (*Phased Array Monitoring for Enhanced Life Assessment*) SHMTM System [1][2] is an integrated embedded system consisting of several electronic devices and one system manager controller. Each PAMELA device is capable of generating up to twelve excitation signals to be applied to an array of integrated piezoelectric Phased Array (PhA) transducers stuck to aircraft structure, acquiring the response signals reflected. Each device is connected with the system manager controller in order to receive the configuration parameters for the desired tests and sending the obtained data. Communication between each device and the system controller may be established by Ethernet interface or by creating a Wi-Fi-based wireless mesh network (WMN) [3].

PAMELA devices consist of several hardware subsystems for generating excitation signals, acquiring data, communicating with the system controller, and so on. A significant part of the hardware system is based on a Virtex 5 FPGA (Field Programmable Gate Array) device from Xilinx with an embedded PowerPC processor running an embedded Linux distribution.

The system can perform several types of tests [1]: passive, simple, round robin, transmitter beamforming, transmitter focusing, time reversal (simple and round robin), and plane front, with selectable wave velocities, sampling frequency and number of samples per test. The excitation signals are fully configurable in shape, frequency, amplitude and number of pulses. The type of excitation signals includes sine, sine sweep, impulse, uniform white noise and arbitrary, with several windowing as Hamming, Hanning, Flat, etc. In addition, it is possible to carry out Pitch-catch tests with multiple SHM systems.

In the classical approach of structural health monitoring systems, the electronic devices involved are in charge of generating the excitation signals, acquiring the response signals and sending the temporal raw data to a computer for further processing. When PAMELA SHM System works in this mode, each PAMELA device sends the raw data acquired to the system manager controller, without any kind of data processing applied.

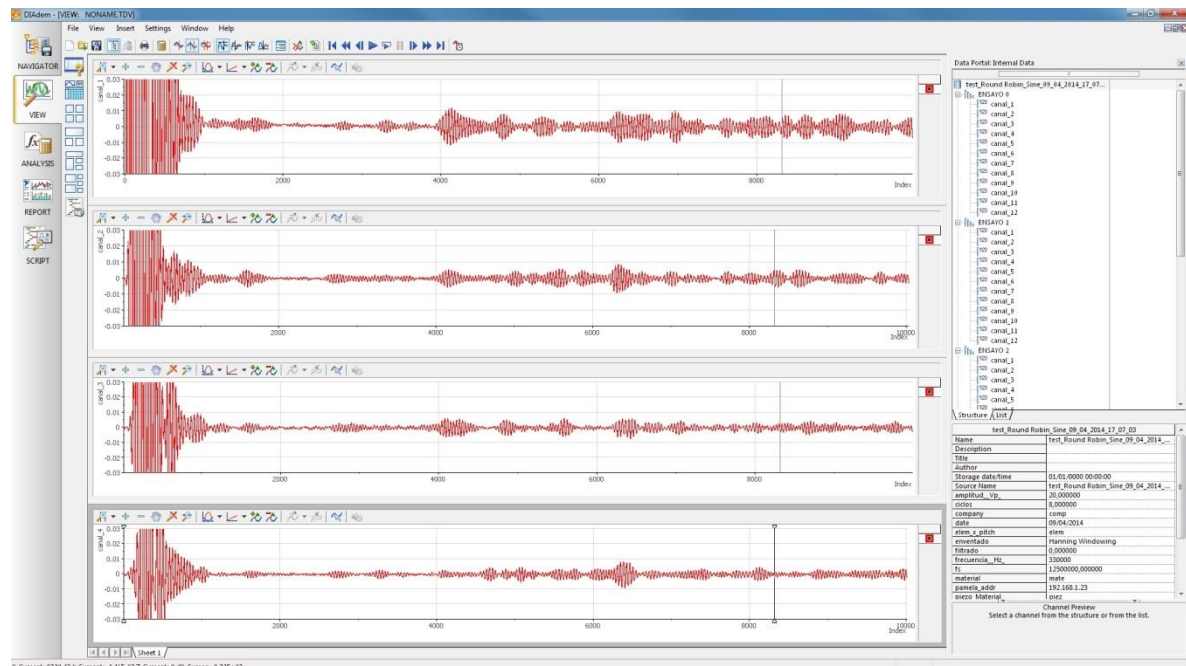


Figure 1. Temporal raw waveforms acquired by PAMELA (DIAdem view)

Figure 1 shows the temporal raw waveforms from several channels acquired by one PAMELA device using a simple test. Using this approach, all data have to be sent from all SHM devices to the central controller, who is in charge of further data processing. This causes high network traffic and high CPU load in the controller, since the amount of data acquired is usually high.

However, a different approach may be considered. As mentioned previously, each PAMELA device integrates a FPGA with an embedded PowerPC processor running an embedded Linux distribution. Therefore, each device could provide local processing capability to the entire system. In this way, each device could perform a set of tests (simple, round-robin, beamforming, etc.), acquiring the data, making data processing or pre-processing (reduction, normalization, pattern recognition, feature extraction, etc.), and sending to the controller only the final results of the structural integrity analysis. This new approach presents several advantages: a) The global processing capacity of the system increases, since each individual device can process locally each own acquired data; b) The data traffic over the mesh network decreases, as only the final results must be sent from each device to the system controller. As an example, in a structural damping versus frequency algorithm (SDFA), with 200-400 kHz frequency sweep in steps of 1 kHz and 25,000 samples per channel, the network traffic is the 0.0055% of the previous one using the classical approach [4]; c) The system controller CPU load decreases, due to the local data processing performed in each device; d) Each PAMELA device can operate in autonomous mode. In this new scenario, each individual device can be configured to carry out a set of specific tests in a scheduled sequence. After the tests, the device can perform the predefined data processing algorithms, for example, to detect structural damage. According to the results achieved, the device can decide to do nothing, sending waveforms and/or results to the controller, signalling an alarm, and so on.

The remainder of the paper is organized as follows: The next section presents the embedded local data processing system, including a description of the different modules in the system and the data flow between them. Section 2 describes the necessary steps for integrating a new data algorithm in the system. Section 3 provides an example of a basic damage index estimation using delay and sum algorithm integrated in the system. Finally, some conclusions are provided.

1 EMBEDDED DATA PROCESSING

Previous section indicates the capability of local data processing of each PAMELA device. This section presents the main components of the system that allow the embedded data processing and the communication between all devices and the system controller.

Figure 2 represents the entire block diagram of the system. The upper section of the figure represents modules belonging to the system manager controller; while the lower part corresponds with subsystems of one PAMELA device (the whole system may integrate hundreds of PAMELA devices).

The system manager controller consists of a computer running Windows OS, and is composed of two differentiated blocks. The left part of the upper section of Figure 2 represents the configuration user interface module (CUIM). This module has been developed in LabVIEW and allows configuring the behaviour of all PAMELA devices. Each device belonging to the system has its own IP address, so the CUIM can connect to the SMA running in each PAMELA and can send commands for performing specific experiments to each individual device. These commands include all the parameters required for the specific experiment (test type, signal generation parameters, type of data processing algorithm, etc.) as well as the scheduling for performing them (an example can be seen in Section 3). This information is organized as an *xml* (*eXtensible Markup Language*) string, where each parameter of the experiment corresponds with an *xml* element. The CUIM includes a TCP/IP client to communicate with PAMELA TCP/IP servers.

The SMA consists of several modules. The first module involved in the data flow from the point of view of PAMELA devices (see lower part of Figure 2) is the PAMELA TCP/IP Server (PS). This module allows the communication with the CUIM in order to receive the *xml* string containing the parameters of a next experiment to be performed by PAMELA hardware. The PS sends the received *xml* string to an *xml* parser to check that the string is well formed according to the rules of XML syntax and verify that the *xml* code is valid. According to the *xml* parser, status information is sent from PS to CUIM.

Once the *xml* string has been verified, its information is inserted, according with their scheduling parameters, in an Experiment Linked List (ELL) managed by the Scheduler Module. The Scheduler Module uses the Linux system clock to control the experiments schedule. Linux Network Time Protocol (NTP) is used to synchronize the Linux system clock with the computer one [4] in order to use the same absolute time in the whole system.

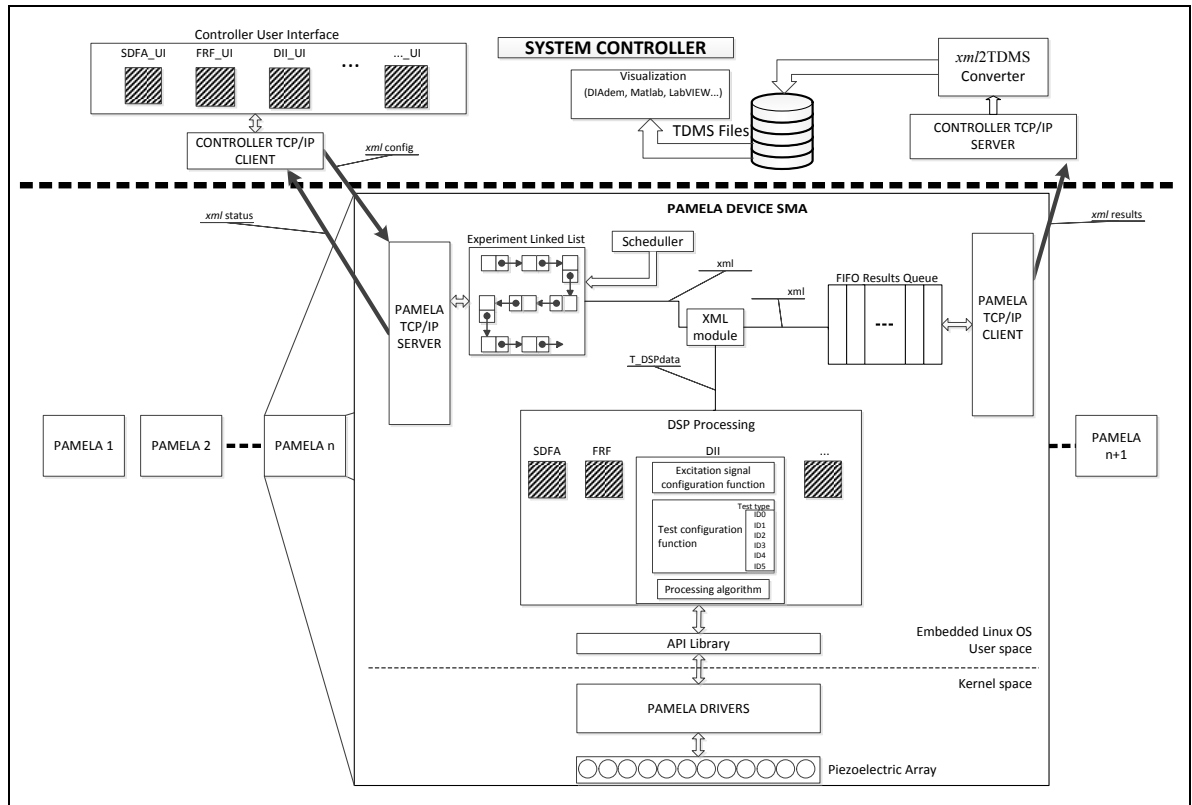


Figure 2. Block diagram of the system

When an experiment's timing has been reached, the Scheduler Module extracts it from ELL and calls the *xml* module to extract the experiment parameters and fills in a C struct ($t_DSPdata$), which includes all the DSP process parameters: a) The excitation signals parameters; b) Number and type of test/s to be performed with all their configuration values; c) The further local data processing algorithm to be applied to the acquired data and its specific configuration.

The DSP Processing Module (DPM) receives the configuration information from $t_DSPdata$ struct and calls the corresponding DSP experiment process function (SDFA, FRF, DII, etc. in Figure 2).

Table 1 shows the test types supported by PAMELA hardware. Different combinations of these types allow carrying out passive tests (base noise level, impact detection, ...), simple test, round-robin, beamforming, focussing, time reversal, etc. Execution of each test type defined in Table 1 generates twelve files (one per piezoelectric transducer) containing the acquired data. The DSP

experiment functions must invoke many times as necessary these test types to reach their goals. For instance, to perform a round-robin, it would be necessary to execute twelve times the test ID0, modifying in each iteration the channel used to generate the excitation.

Table 1: Test types supported.

Test ID	# signals	# BSMs	# excitation channels	Delay between signals	Test type
ID0	1	1	1	no	Simple Round-robin
ID1	1	2-12	1	no	
ID2	12	12	12	no	Plane-front
ID3	1	1	12	yes	Beamforming
ID4	1	2-12	12	yes	Focusing
ID5	0	NA	NA	NA	Passive

All the DSP process functions must be codified to implement these tasks: a) Set the number of tests to do and configure each test with the excitation signals parameters and the test configuration defined in the user configuration struct; b) Execute the test in the device (from those defined in Table 1); c) Apply the local data processing algorithm to the test results stored in the result files (PAMELA devices use a RAM-based filesystem); d) Send to the *xml* module the *t_DSPdata* struct. The DSP process functions use the API library in user space to configure the hardware operations. This API library uses the functions of the Linux PAMELA driver to manage the hardware operations. The *xml* module sends the *xml* string with the results of the data processing algorithms, and maintaining the configuration information previously stored, to the FIFO Results Queue.

PAMELA TCP/IP Client (PC) is in charge of extracting the results stored in the FIFO and sending them to the System Controller TCP/IP Server. This will be the last step in the data flow at PAMELA device level.

The right part of the upper section of Figure 2 shows the System Controller TCP/IP Server that receives the *xml* string containing both the test's configuration information and the obtained results. The *xml2TDMS* Module extracts all the data from *xml* string and creates a TDMS (Technical Data Management Streaming) file to be stored in a local database or in a cloud server. TDMS files are structured using three levels of hierarchy (file, groups, and channels) and allow storing in a single file all data attributes together with the data itself. The information from TDMS files can be retrieved for exploitation using LabVIEW, Excel, DIAdem, Matlab, etc. Each time a TDMS file is created, the *xml2TDMS* Module update the experiment status to identify it has finished.

2 STEPS FOR NEW ALGORITHMS INTEGRATION

The software architecture presented in the previous section has been designed to allow integrating new data processing algorithms in the system in an easy, cost-effective and timely way. To create new experiments and its associated data processing algorithms is not necessary to modify the general software architecture, but merely to codify the new modules and to integrate them in the system.

Figure 2 shows some of the modules shaded with oblique stripes (SDFA, FRF, DII, etc.). These modules are the only ones to be added or modified, while the rest of the systems will remain unchanged.

2.1 Changes in the system controller

From the point of view of the system controller, only the CUI must be modified to integrate a new type of experiment. It will be necessary to register the new type in the CUI list of experiments, to develop the new user interface (LabVIEW based) including all the required parameters to configure it, and to codify, also in LabVIEW, a simple application to manage the user

experiment and integrate it with the rest of the CUIM. For this purpose, a template is offered to simplify the new experiment development.

2.2 Changes in PAMELA devices

Several steps must be taken in PAMELA devices to add data processing algorithms to the SMA. First of all, it is necessary to codify the new DSP process function to implement the new experiment. The function must implement the tasks described in the embedded data processing section. In addition, it is necessary to add in the xml module the code necessary to manage the new experiment parameters.

With the aim to simplify as far as possible the integration of new experiments in the system, a virtual machine (VMware based) with an Ubuntu 12.04 Linux distribution is offered. This virtual machine integrates an Eclipse IDE completely configured to meet the system requirements. Within Eclipse, a template with the SMA project containing an example of one DSP experiment function is provided. Based on the example offered, a new DSP experiment function must be codified. Finally, the application must be cross compiled (using integrated Eclipse-PowerPC cross compiler) and downloaded (copied) to the PAMELA filesystem. This last process is fully transparent for the user, as the virtual machine is previously configured for this.

3 BASIC DAMAGE INDEX ESTIMATION ALGORITHM

This section describes the integration of a basic experiment in the system to calculate the damage index estimation using delay and sum algorithm.

Basic damage index estimation (BDIE) is performed using a round-robin test, and taking into account the contributions of each emitter–receiver pair using the delay and sum procedure. As the experiment is based on a round-robin test, the input parameters refer only to the excitation signal values and to the inspection area coordinates. Figure 3 shows the experiment's user interface, where *Test Info* and *schedule* parameters are common to all experiments and *BDIE parameters* are the specific parameters of the experiment.

Test Info	BDIE Parameters	schedule
Test Name: Damage Index Image	Amplitude[Vp]: 200000	Type: Now
Company: Technical University of Madrid	Frequency[Hz]: 330000	Time: 12:10:25,970
Prototype: PNJ345-87	Cycles: 8	11/04/2014
Material: Aluminum	Nsamples: 25000	
PhA Transducer: PhA SN 0000015	WINDOW_FORM: Hanning	
Condition/state: PostDamage	Xstart: -5	
Piezo Material: 3G765F	Xend: 15	
IP: 192.168.1.23	x_accuracy: 2	
MAC: 03-98-A7-3D-F3-27	Ystart: 25	
	Yend: 45	
	y_accuracy: 2	
START		

Figure 3. User interface for basic damage index estimation algorithm

Based on the information of the experiment's user interface, the CUIM will compose an *xml* string (see Figure 4) that will be sent to the PAMELA TCP/IP Server. The experiment needs to perform a round-robin test, iterating up to 12 ID0 test (see Table 1), using in each iteration one

different channel to generate the excitation. As a result of this process, 144 files of 25,000 samples each one will be generated. These data will be the inputs of the implemented damage index estimation algorithm. The algorithm is described in detail in [5].

<TEST>	<Type>String</Type>
<INFO>	<Valor>Hanning Windowwing</Valor>
<Testname>Damage Index Image</Testname>	</Parameters>
<Company>Technical University of Madrid</Company>	<Parameters>
<Prototype>PNJ345-87</Prototype>	<Name>Xstart</Name>
<Material>Composite</Material>	<Type>SGL</Type>
<PhA_Transducer>PhA SN 0000015</PhA_Transducer>	<Valor>-5,000000</Valor>
<Conditionstate>PostDamage</Conditionstate>	</Parameters>
<Piezo_Material>3G765F</Piezo_Material>	<Parameters>
<IP>192.168.1.23</IP>	<Name>Xend</Name>
<MAC>03-98-A7-3D-F3-27</MAC>	<Type>SGL</Type>
</INFO>	<Valor>15,000000</Valor>
<SCHEDULE>	</Parameters>
<Type>Now</Type>	<Parameters>
<Time>2014-04-11T11:59:20,267+01:00</Time>	<Name>x_accuracy</Name>
</SCHEDULE>	<Type>SGL</Type>
<PARAMETERS>	<Valor>2,000000</Valor>
<Name>FRF</Name>	</Parameters>
<Parameters>	<Parameters>
<Name>Amplitude[Vp]</Name>	<Name>Ystart</Name>
<Type>SGL</Type>	<Type>SGL</Type>
<Valor>200000,000000</Valor>	<Valor>25,000000</Valor>
</Parameters>	</Parameters>
<Parameters>	<Parameters>
<Name>Frequency[Hz]</Name>	<Name>Yend</Name>
<Type>SGL</Type>	<Type>SGL</Type>
<Valor>330000,000000</Valor>	<Valor>45,000000</Valor>
</Parameters>	</Parameters>
<Parameters>	<Parameters>
<Name>Cycles</Name>	<Name>y_accuracy</Name>
<Type>SGL</Type>	<Type>SGL</Type>
<Valor>8,000000</Valor>	<Valor>2,000000</Valor>
</Parameters>	</Parameters>
<Parameters>	</PARAMETERS>
<Name>Nsamples</Name>	<RESULT/>
<Type>SGL</Type>	<STATUS>
<Valor>25000,000000</Valor>	<error1>0</error1>
</Parameters>	<error2>0</error2>
<Parameters>	</STATUS>
<Name>WINDOW_FORM</Name>	</TEST>

Figure 4. *xml* string sent by the system controller to PAMELA

In this example, the implemented algorithm does not consider the dispersion compensation. Therefore the damage index estimation for each observation point $O(x_o, y_o)$ of the grid is calculated using Equation (1), where N is the number of piezoelectric transducers, R and E represent receiver and emitter respectively, $H(S_{ER}(t))$ is the complex representation of the signal using the Hilbert transform and $T_{ER}^1(O, t)$ is the theoretical signal associated to the observation point O (see [5] for additional details about each term in this equation).

$$D_1(O) = \left| \sum_{E=1}^N \sum_{R \neq E} \int_{-\infty}^{+\infty} H(S_{ER}(t)) T_{ER}^1(O, t) dt \right| \quad (1)$$

The result of the algorithm will be a 2D array with the damage index estimation assigned to each observation point of the grid.

Figure 5 represents the obtained results for two different experiments performed. Figure 5a shows the specimen used to carry out the tests. The lower part of the figure shows one PAMELA device connected to the piezoelectric transducer (below it) bonded to the aluminium plate. In the upper part the inspection area is marked with a red square. Inside the inspection area, A and B damages has been marked with blue circumferences. Figure 5b is the 2D representation of the

calculated damage index when the specimen only presents the A damage, while Figure 5c is the 2D representation of the calculated damage index when the specimen presents both the A and B damages.

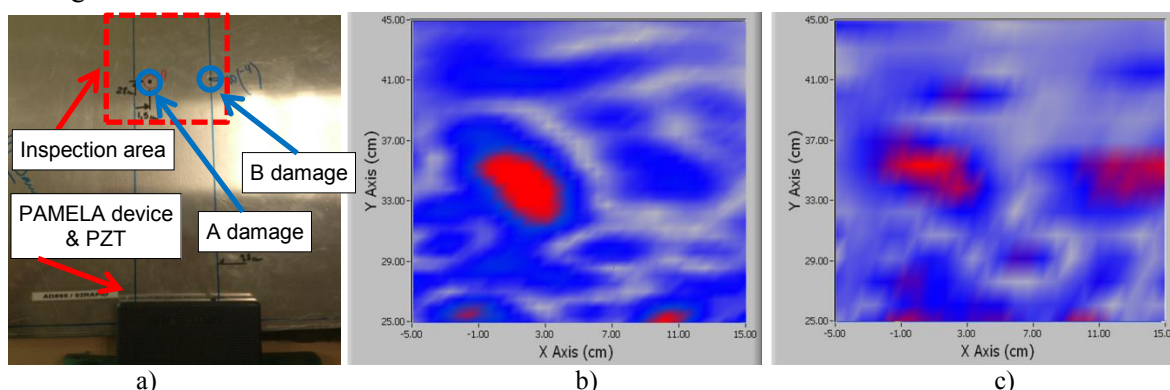


Figure 5. a) specimen used to carry out the tests; b) 2D representation of damage index (A damage); c) 2D representation of damage index (A and B damages)

Table 2 shows the amount of data (samples) generated by the algorithm and its comparison with the amount of data (samples) of the raw waveforms for an inspection area of $20 \times 20 \text{ cm}^2$ for different grid density (accuracy indicates the points per cm using to form the grid).

Table 2: Data reduction.

Accuracy (points per cm)	Amount of data (samples)	Data reduction respect raw data (%)
1	400	0.01
2	1,600	0.04
5	10,000	0.27
10	40,000	1.11

CONCLUSION

The paper describes PAMELA SHMTM System as a fully integrated system for traditional SHM applications, and focuses later its attention in a novelty approach based on the capability of PAMELA devices to perform local data processing. This new approach provides multiple advantages, as described in the paper. Well-defined software architecture allows an extremely easy and quick way to integrate new data processing algorithms in the system. These features and the interesting results obtained using a very simple and basic damage index estimation algorithm make the system very attractive for all kind of SHM applications.

REFERENCES

- [1] P.M. Monje et al. Integrated Electronic System for Ultrasonic Structural Health Monitoring. *6th European Workshop on Structural Health Monitoring - Fr.1.C.1*, 2012.
- [2] G. Aranguren et al. Ultrasonic wave-based structural health monitoring embedded instrument. *Review of Scientific Instruments* 84, 125106 (2013); doi: 10.1063/1.4834175
- [3] E. Barrera et al. Structural Health Monitoring Network System with Wireless Communications Inside Closed Aerospace Structures. *6th European Workshop on Structural Health Monitoring - Tu.3.A.3*, July 2012.
- [4] E. Barrera et al. Multiple PAMELA SHMTM system for automated SHM control during fatigue tests of aircraft structures. *9th International Workshop on Structural Health Monitoring (Stanford University, USA)*, 2013.
- [5] N Quaegebeur, P C Ostiguy and P Masson. Correlation-based imaging technique for fatigue monitoring of riveted lap-joint structure. *Smart Materials and Structures* 23 (2014) 055007 (10pp) doi:10.1088/0964-1726/23/5/055007